**Octave/Matlab Tutorial**

1.Suppose I first execute the following Octave/Matlab commands:

A = [1 2; 3 4; 5 6];

B = [1 2 3; 4 5 6];

Which of the following are then valid commands? Check all that apply. (Hint: A' denotes the transpose of A.)

C = A \* B;

C = B' + A;

C = A' \* B;

C = B + A;

2.Let *A*=⎢16594211714310615138121⎤⎦⎥⎥.

Which of the following indexing expressions gives *B*=⎡⎣⎢⎢16594211714⎤⎦⎥⎥? Check all that apply.

B = A(:, 1:2);

B = A(1:4, 1:2);

B = A(0:2, 0:4)

B = A(1:2, 1:4);

3.Let *A* be a 10x10 matrix and *x* be a 10-element vector. Your friend wants to compute the product *Ax* and writes the following code:

v = zeros(10, 1);

for i = 1:10

for j = 1:10

v(i) = v(i) + A(i, j) \* x(j);

end

end

How would you vectorize this code to run without any FOR loops? Check all that apply.

v = A \* x;

v = Ax;

v = A .\* x;

v = sum (A \* x);

4.Say you have two column vectors *v* and *w*, each with 7 elements (i.e., they have dimensions 7x1). Consider the following code:

z = 0;

for i = 1:7

z = z + v(i) \* w(i)

end

Which of the following vectorizations correctly compute z? Check all that apply.

z = sum (v .\* w);

z = w' \* v;

z = v \* w;

z = w \* v;

5.In Octave/Matlab, many functions work on single numbers, vectors, and matrices. For example, the sin function when applied to a matrix will return a new matrix with the sin of each element. But you have to be careful, as certain functions have different behavior. Suppose you have an 7x7 matrix *X*. You want to compute the log of every element, the square of every element, add 1 to every element, and divide every element by 4. You will store the results in four matrices, A, B, C, D. One way to do so is the following code:

for i = 1:7

for j = 1:7

A(i, j) = log(X(i, j));

B(i, j) = X(i, j) ^ 2;

C(i, j) = X(i, j) + 1;

D(i, j) = X(i, j) / 4;

end

end

Which of the following correctly compute A, B, C or D? Check all that apply.

C = X + 1;

D = X / 4;

B = X .^ 2;

B = X ^ 2;